

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний інститут  
імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 6 з дисципліни  
«Алгоритми та структури даних-1.  
Основи алгоритмізації»

«Дослідження рекурсивних

алгоритмів»

Варіант 3

Виконав студент

Баран Софія Володимирівна  
(шифр, прізвище, ім'я, по батькові)

Перевірів

( прізвище, ім'я, по батькові)

# Лабораторна робота 6

## Дослідження рекурсивних алгоритмів

**Мета** – дослідити особливості роботи складних циклів та набутти практичних навичок їх використання під час складання програмних специфікацій.

Варіант-3

### 1. Завдання:

3. Обчислення суми 10 елементів арифметичної прогресії, що зростає: початкове значення – 10, крок – 2

2. **Постановка задачі:** Для знаходження суми арифметичної прогресії скористаємося підпрограмою, яка працює з трьома аргументами: к-стю елементів, першим елементом та різницею прогресії. В тілі підпрограми буде відбуватися рекурсія. Умова виходу з рекурсії: к-сть елементів = 1.

### Математична модель:

Змінна	Тип	Ім'я	Призначення
Перша змінна	Цілочисельний	n	Початкове дане
Друга змінна	Цілочисельний	first_elem	Початкове дане
Третя змінна	Цілочисельний	diff	Початкове дане
Четверта змінна	Цілочисельний	elem	Проміжні дані
П'ята змінна	Цілочисельний	res	Результат

Формули, що використовуються:

- $elem = first\_elem + diff * (n - 1)$

### 3. Псевдокод:

*Крок 1:* Визначимо основні дії

*Крок 2:* Деталізуємо умову з використанням підпрограми та умовних операторів.

#### **Крок 1:**

##### **Початок**

n = 10

first\_elem = 10

diff = 2

Визначення суми арифметичної прогресії за допомогою підпрограми;

Виведення res;

##### **Кінець**

#### **Крок 2:**

##### **Підпрограма**

sum\_of\_progression(n, first\_elem, diff):

elem = first\_elem + diff \* (n - 1)

**якщо** n == 1:

**повернути** first\_elem

**інакше**

**повернути** elem + sum\_of\_progression(n - 1, first\_elem, diff)

**все якщо**

##### **Кінець підпрограми**

##### **Початок**

n = 10

first\_elem = 10

diff = 2

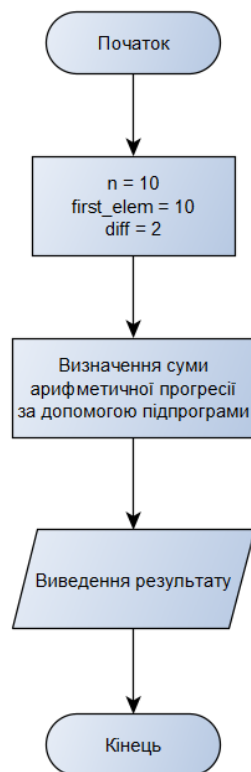
res = sum\_of\_progression(n, first\_elem, diff)

вивести res

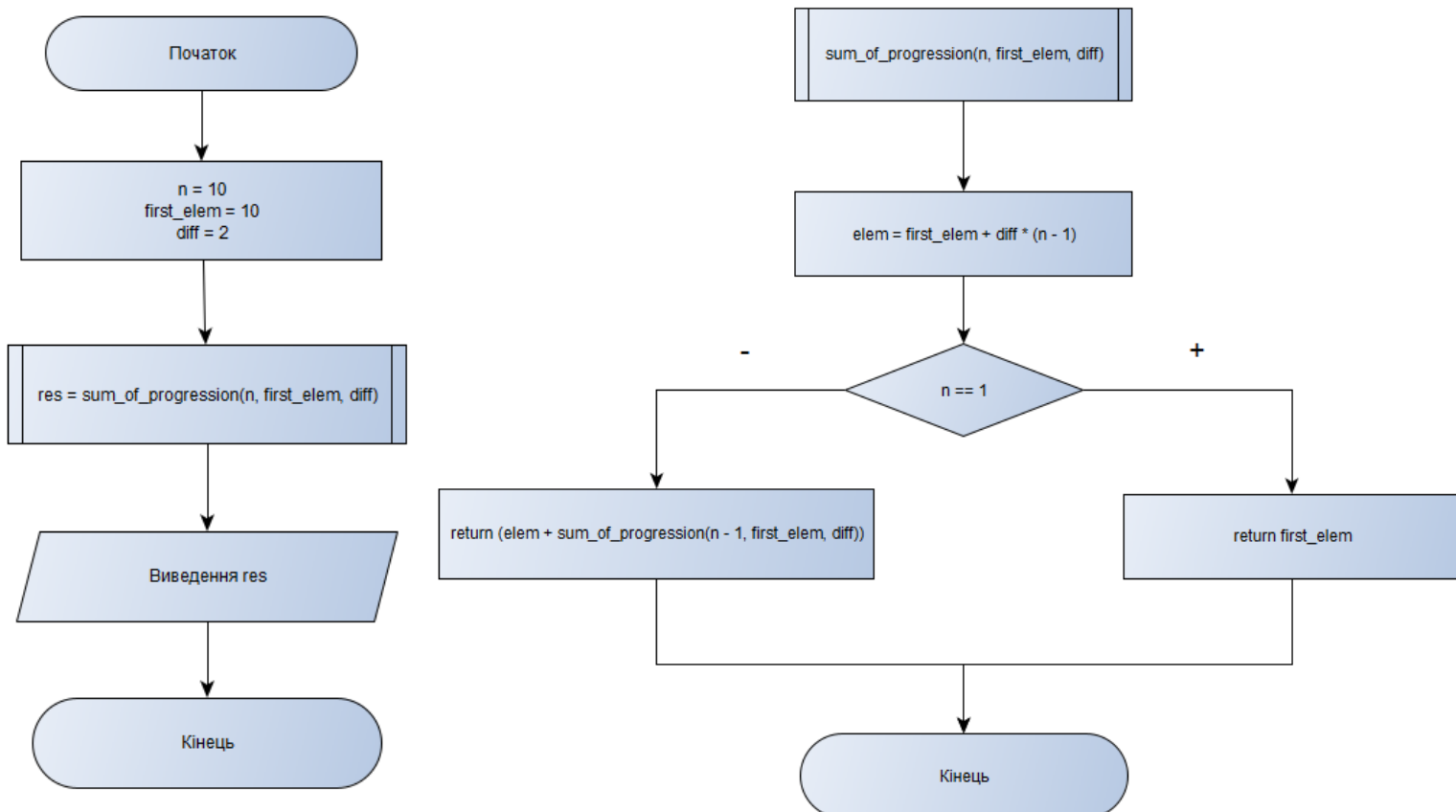
##### **Кінець**

## 4. Блок схеми

### Крок 1



### Крок 2



## 5. Код програми на мові python:

```
from os import system
```

```
def sum_of_progression(n, first_elem, diff):
```

```
    elem = first_elem + diff * (n - 1)
```

```
    if n == 1:
```

```
        return first_elem
```

```
    else:
```

```
        return elem + sum_of_progression(n - 1, first_elem, diff)
```

```
def main():
```

```
    n = 10
```

```
    first_elem = 10
```

```
    diff = 2
```

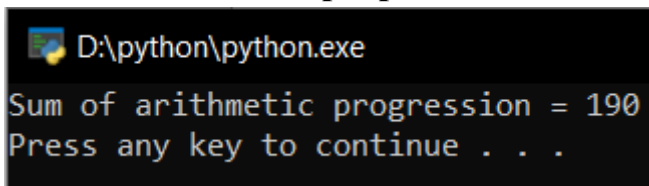
```
    res = sum_of_progression(n, first_elem, diff)
```

```
    print("Sum of arithmetic progression =", res)
```

```
main()
```

```
system("pause")
```

## 6. Виконання програми:



```
D:\python\python.exe
Sum of arithmetic progression = 190
Press any key to continue . . .
```

## **7. Висновок:**

На лабораторній роботі я набула навички роботи з підпрограмами та набула практичних навичок їх використання під час складання рекурсивних алгоритмів. У результаті лабораторної роботи було розроблено математичну модель, що відповідає постановці задачі; псевдокод, блок-схеми та код на мові python, які пояснюють логіку алгоритму. Використовуючи рекурсивну підпрограму було визначено, що сума арифметичної прогресії, за заданими в умові значеннями,  $= 190$ .